Lab 0 Report

ECE 332 Winter 2020

Jose Manuel Lopez Alcala & Benjamin Carlson

1. Introduction

In this first lab, we learned about dSpace, Simulink, and compiling our design to the microcontroller. In addition to this, we were able to read data from the encoder on the motor and graph the results using dSpace software. Overall, this lab helped us get familiar with equipment and software that will be used in the following labs.

2. Methods

2.1. Connecting the Hardware

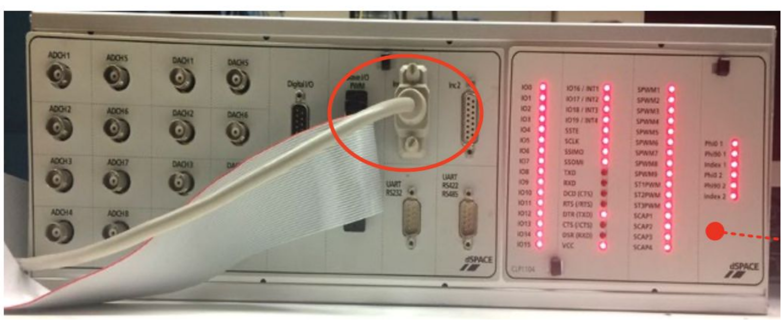

Figure1. Encoder connected to box for communication

As shown in Figure1, the motor's encoder must be connected to the grey box so that data can be obtained. This connection is labeled as Inc1, but it can be connected to Inc2 as well. The only thing to keep in mind is that this matters when coding up the design.

The second thing to setup is the motor. The motor needs to be connected to power using connections A and B shown in Figure 2. These connectors get their power from the power supply shown in Figure 3.

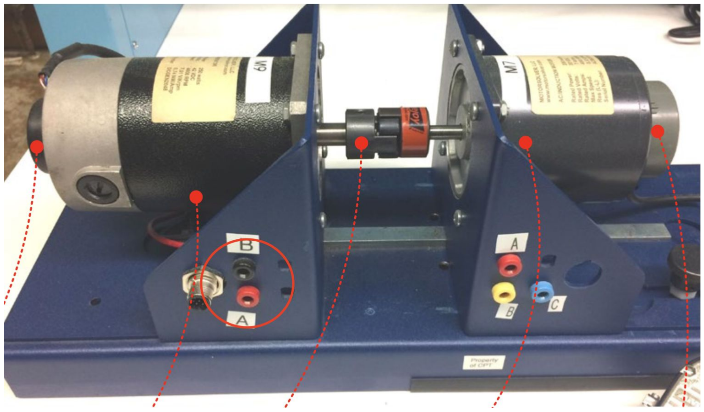

Figure2. The motor setup

Depending on the voltage set, the motor will spin faster or slower. It is best to have the power supply turned off so that the motor is not running while it is not supposed to.

Figure3. The power supply used for motor

2.2. Programming using Simulink & dSpace

Using Simulink, we created a new project using the dSpace 1104 library. We then modeled the encoder to input through channel 1 of the box, which monitors the motor's angular position. To do this, we constructed a block to accept one input, the position, and produces three outputs. One being the change in position, the other two being the angular velocity in radians per second and RPM's. Knowing that the encoder pulses 1000 times per 2π radians of rotation, we multiplied the channel 1 input by $2\pi/1000$, which gave us the change in angular position for any time(t). To find the angular velocity in rad/s, we created another gain stage to multiply the output by the change in position as above and divide by a constant time step, which will be determined later on. Finally, the third output determines the velocity in RPM by multiplying output 2(angular velocity in rad/s) by 60/2π.

After the diagram was finished, we exported the design using the Code Generation feature in Simulink. Once that was complete, we configured several parameters to ensure proper graph generation. Specifically, the Solver type to “fixed” and selecting the ODE4 graphing tool to display our graphs. We then configured the time step to be a constant 0.1 ms before compiling and deploying our program.

3. Results

After uploading our designs to the board, we were able to retrieve data using dSpace. We retrieved angular velocity, position, and RPM. The following images present our data at different voltages supplied to the motor: 5, 10, 15, 20 Volts.

In Figure 4 it can be seen from the graphs that the angular velocity reaches stability at 0.63E3 radians per second and RPM stabilized at 0.60E4. The motor had 5 volts applied to it.

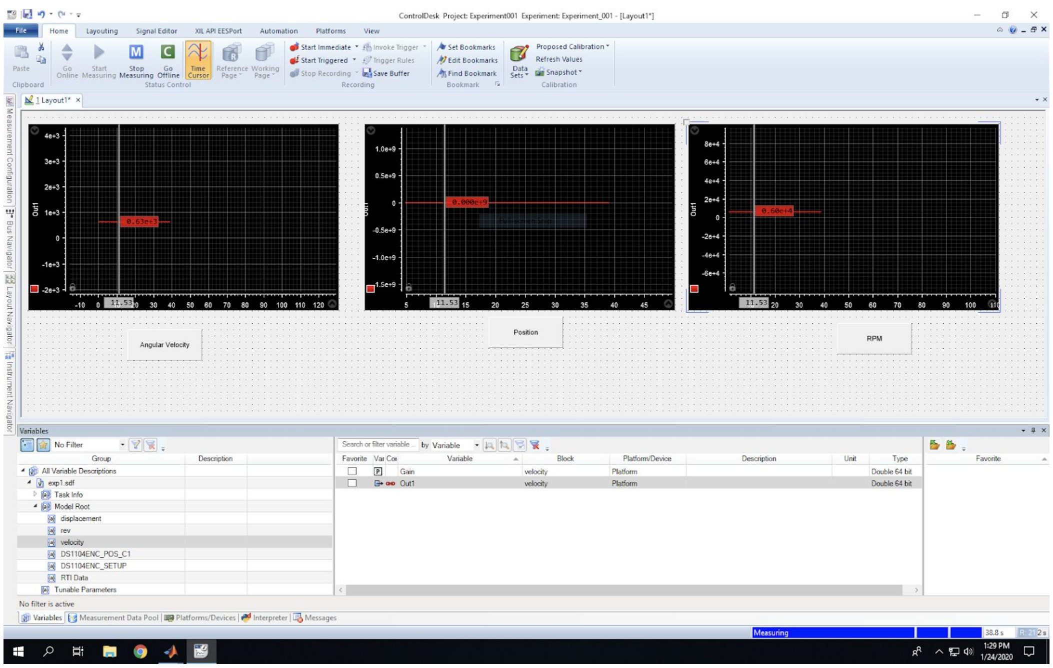

Figure 4. Angular velocity, position and RPM graph when the motor was running at 5 Volts.

Figure 5 shows that the angular velocity tapers off at 1.398E3 and RPM stabilize at 0.134E5. The motor had 10 volts applied to it.

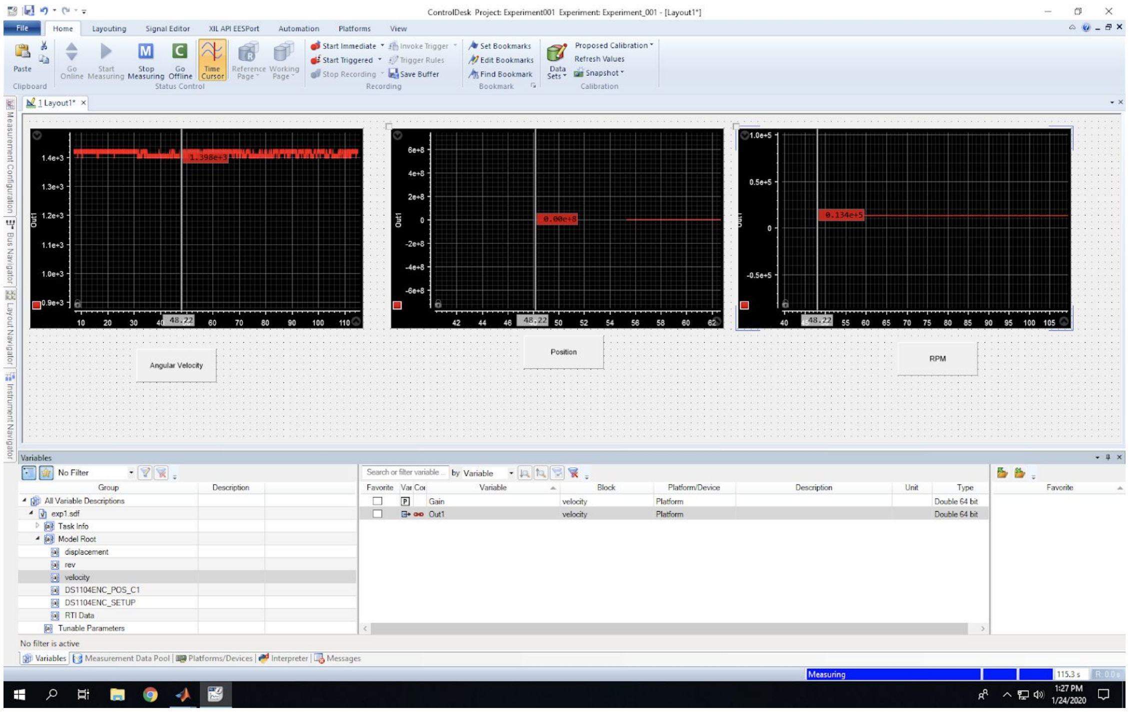

Figure 5. Angular velocity, position and RPM graph when the motor was running at 10 Volts.

Figure 6 shows that the angular velocity tappers off at 2.09E3 radians/second and the RPM at 2.00E4. The motor had a supply of 15 volts.

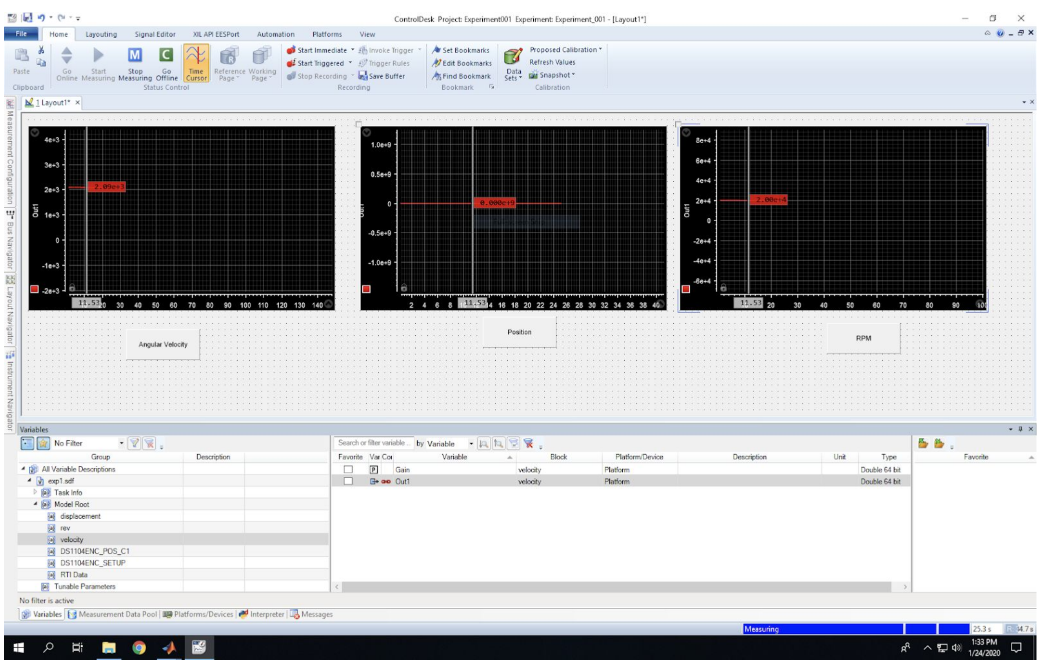

Figure 6. Angular velocity, position and RPM graph when the motor was running at 15 Volts.

Lastly, Figure 7 shows the motor operating at 20 volts. The angular velocity stabilizes at around 2.81E3 radians/second and at 2.69E4 for RPM.

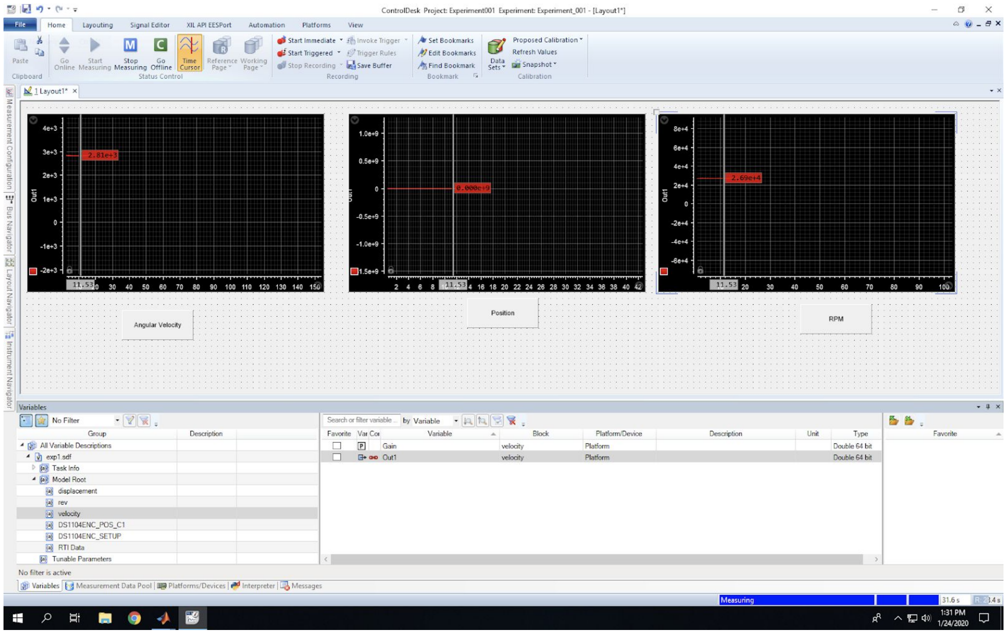

Figure 7. Angular velocity, position and RPM graph when the motor was running at 20 Volts.

4. Discussion

When running the motor at higher voltages it was observable that the motor was operating at a higher speed and we could verify this by looking at the graphs that were generated from dSpace. This is observable in both the RPM and the angular velocity. Although it was not captured, the position graph showed that the position keeps increasing as long as the motor keeps on rotating.

The setup of the experiment was fairly simple. It involved connecting the encoder to the gray box, designing the parameters of the system, and uploading the program to the controller. This lab gave a good overview of all of the basic tools that will be needed in order to complete future labs.